Excelente! Desarrollar tu propio CMS desde cero utilizando **JavaScript**, **HTML5**, **CSS3**, **React** y **MongoDB** es un proyecto ambicioso y muy interesante. Esto te dará un control total sobre el diseño, la funcionalidad y la escalabilidad del sitio web de tu iglesia. A continuación, te guiaré paso a paso para que puedas lograrlo:

**1. Planificación del Proyecto**

Antes de empezar a codificar, es importante planificar:

* **Requisitos del sitio web:**
  + Páginas estáticas (Inicio, Sobre nosotros, Contacto, etc.).
  + Gestión de eventos (calendario, creación, edición y eliminación).
  + Publicación de sermones (texto, audio, video).
  + Sistema de donaciones en línea.
  + Autenticación de usuarios (administradores, miembros).
  + Panel de administración para gestionar contenido.
* **Tecnologías:**
  + **Frontend:** React (para la interfaz de usuario), HTML5, CSS3.
  + **Backend:** Node.js con Express (para la API).
  + **Base de datos:** MongoDB (para almacenar datos como usuarios, eventos, sermones, etc.).
  + **Autenticación:** JWT (JSON Web Tokens) para manejar sesiones de usuarios.

**2. Estructura del Proyecto**

Organiza tu proyecto en dos partes principales: **frontend** y **backend**.

**Frontend (React)**

* **Componentes principales:**
  + Header: Barra de navegación.
  + Footer: Información de contacto y enlaces.
  + HomePage: Página de inicio con mensajes destacados.
  + AboutPage: Información sobre la iglesia.
  + EventsPage: Lista de eventos con un calendario.
  + SermonsPage: Archivo de sermones.
  + DonationsPage: Formulario de donaciones.
  + AdminPanel: Panel de control para gestionar contenido (accesible solo para administradores).
* **Herramientas útiles:**
  + React Router: Para manejar la navegación entre páginas.
  + Axios: Para hacer solicitudes HTTP a tu backend.
  + Material-UI o TailwindCSS: Para estilizar componentes de manera rápida y profesional.

**Backend (Node.js + Express + MongoDB)**

* **Endpoints de la API:**
  + GET /api/events: Obtener lista de eventos.
  + POST /api/events: Crear un nuevo evento (protegido por autenticación).
  + GET /api/sermons: Obtener lista de sermones.
  + POST /api/sermons: Subir un nuevo sermón (protegido por autenticación).
  + POST /api/donations: Procesar una donación.
  + POST /api/auth/login: Autenticar usuarios.
  + POST /api/auth/register: Registrar nuevos usuarios (administradores).
* **Base de datos (MongoDB):**
  + **Colección users:** Almacena información de usuarios (nombre, email, contraseña, rol).
  + **Colección events:** Almacena eventos (título, descripción, fecha, lugar).
  + **Colección sermons:** Almacena sermones (título, descripción, archivo de audio/video, fecha).
  + **Colección donations:** Almacena registros de donaciones (monto, donante, fecha).

**3. Desarrollo del Backend**

**Configuración inicial:**

1. Instala Node.js y MongoDB en tu sistema.
2. Crea un nuevo proyecto con npm init.
3. Instala las dependencias necesarias:

npm install express mongoose cors jsonwebtoken bcryptjs

**Estructura del backend:**

backend/

├── models/

│ ├── User.js

│ ├── Event.js

│ ├── Sermon.js

│ └── Donation.js

├── controllers/

│ ├── authController.js

│ ├── eventController.js

│ ├── sermonController.js

│ └── donationController.js

├── routes/

│ ├── authRoutes.js

│ ├── eventRoutes.js

│ ├── sermonRoutes.js

│ └── donationRoutes.js

├── middleware/

│ └── authMiddleware.js

└── server.js

**Ejemplo de código para server.js:**

const express = require('express');

const mongoose = require('mongoose');

const cors = require('cors');

const authRoutes = require('./routes/authRoutes');

const eventRoutes = require('./routes/eventRoutes');

const app = express();

app.use(cors());

app.use(express.json());

// Conexión a MongoDB

mongoose.connect('mongodb://localhost:27017/iglesia', {

useNewUrlParser: true,

useUnifiedTopology: true,

});

// Rutas

app.use('/api/auth', authRoutes);

app.use('/api/events', eventRoutes);

const PORT = 5000;

app.listen(PORT, () => {

console.log(`Servidor corriendo en http://localhost:${PORT}`);

});

**4. Desarrollo del Frontend**

**Configuración inicial:**

1. Crea una aplicación React con create-react-app:

npx create-react-app frontend

cd frontend

1. Instala las dependencias necesarias:

npm install axios react-router-dom @mui/material @emotion/react @emotion/styled

**Estructura del frontend:**

frontend/

├── src/

│ ├── components/

│ │ ├── Header.js

│ │ ├── Footer.js

│ │ ├── EventList.js

│ │ └── SermonList.js

│ ├── pages/

│ │ ├── HomePage.js

│ │ ├── AboutPage.js

│ │ ├── EventsPage.js

│ │ ├── SermonsPage.js

│ │ └── DonationsPage.js

│ ├── App.js

│ └── index.js

**Ejemplo de código para App.js:**

import React from 'react';

import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';

import HomePage from './pages/HomePage';

import EventsPage from './pages/EventsPage';

import SermonsPage from './pages/SermonsPage';

import DonationsPage from './pages/DonationsPage';

import Header from './components/Header';

import Footer from './components/Footer';

function App() {

return (

<Router>

<Header />

<Routes>

<Route path="/" element={<HomePage />} />

<Route path="/events" element={<EventsPage />} />

<Route path="/sermons" element={<SermonsPage />} />

<Route path="/donations" element={<DonationsPage />} />

</Routes>

<Footer />

</Router>

);

}

export default App;

**5. Despliegue**

* **Frontend:** Puedes desplegar tu aplicación React en servicios como Vercel, Netlify o GitHub Pages.
* **Backend:** Despliega tu API en servicios como Heroku, Render o DigitalOcean.
* **Base de datos:** MongoDB Atlas es una excelente opción para alojar tu base de datos en la nube.

**6. Mantenimiento y Mejoras**

* Asegúrate de implementar seguridad básica (validación de entradas, protección contra inyecciones SQL, etc.).
* Agrega pruebas unitarias y de integración.
* Escala el proyecto según las necesidades de tu iglesia.

Si necesitas más detalles sobre alguna parte del desarrollo (por ejemplo, cómo implementar la autenticación con JWT o cómo conectar React con tu API), no dudes en preguntar. ¡Estoy aquí para ayudarte en cada paso del camino! 🚀